μTeaboard/ARM7-AT91
取扱説明書

1.00.00
2013年1月

目次

【修正履歴】

Version 1.00.00

・μTeaboard/ARM7-AT91 GNU開発環境(Windows版)説明書のインストール関連部分を分割し、以下のように分冊化。
  ・Cygwin インストール方法説明書(PDF)
  ・Cygwin 用μTeaboard/ARM7-AT91 開発環境インストール方法説明書(PDF)
  ・μTeaboard/ARM7-AT91 GNU開発環境(Windows版)説明書(PDF)

・製品マニュアルに以下を追加。
  ・Eclipse インストール方法説明書(PDF)
  ・Eclipse 用 T-Kernel 開発環境インストール方法説明書(PDF)
  ・GNU開発環境(Eclipse版)説明書(PDF)

Version 1.A0.00

・新規

1. はじめに

1.1 製品構成

μTeaboard/ARM7-AT91 には以下の内容が含まれています。

○ μTeaboard/ARM7-AT91 ハードウェア

詳細は「ハードウェア仕様書」を参照してください。

○ 製品マニュアル

μTeaboard/ARM7-AT91取扱説明書(PDF)
製品の内容および操作方法に関する説明書です。

ライブラリ説明書(PDF)
製品に含まれるC言語ライブラリに関する説明書です。

デバイスドライバ説明書(PDF)
製品に含まれるデバイスドライバに関する説明書です。

TCP/IPマネージャ説明書(HTML)
製品に含まれる TCP/IP マネージャに関する説明書です。

μTeaboard/ARM7-AT91 GNU開発環境説明書(PDF)
製品に含まれる Linux 上で動作する GNU 開発環境のインストール方法および操作方法に関する説明書です。

Cygwin インストール方法説明書(PDF)
GNU開発環境(Windows版)で必要となる Cygwin のインストール方法に関する説明書です。

Cygwin 用μTeaboard/ARM7-AT91 開発環境インストール方法説明書(PDF)
μTeaboard/ARM7-AT91 GNU開発環境(Windows版)のインストール方法に関する説明書です。

μTeaboard/ARM7-AT91 GNU開発環境(Windows版)説明書(PDF)
製品に含まれる Windows上の Cygwin 環境で動作する GNU 開発環境の操作方法に関する説明書です。

Eclipse インストール方法説明書(PDF)
GNU開発環境(Eclipse版)で必要となる、Eclipse と Java のインストール方法に関する説明書です。

Eclipse 用 T-Kernel 開発環境インストール方法説明書(PDF)
GNU開発環境(Eclipse版)のインストール方法に関する説明書です。

GNU開発環境(Eclipse版)説明書(PDF)

製品に含まれる Windows上の Eclipse 環境で動作する GNU 開発環境の操作方法に関する説明書です。

μTeaboard/ARM7-AT91 ハードウェア仕様書
μTeaboard/ARM7-AT91 のハードウェアの仕様書です。

○ 仕様書

T-Monitor 仕様書 (PDF)
T-Monitor の公式仕様書です。

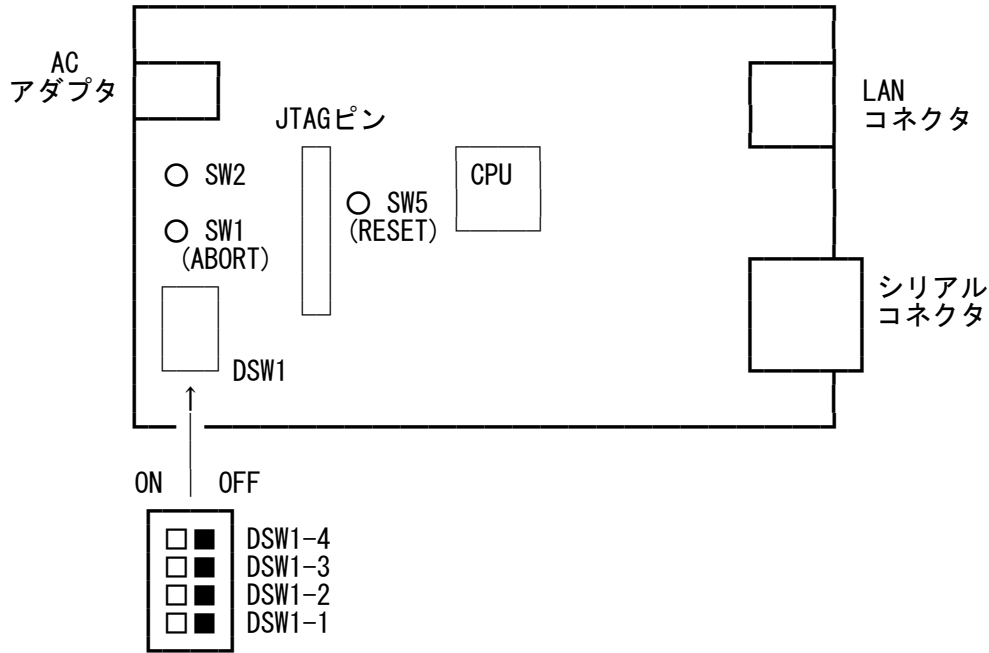
T-Kernel 仕様書 (PDF)
T-Kernel の公式仕様書です。

実装仕様書 (PDF)
T-Monitor/T-Kernel の μTeaboard/ARM7-AT91 の実装に依存した部分の詳細な仕様書です。

○ ソフトウェア
μTeaboard のフラッシュメモリイメージ、GNU の開発環境、ドライバソースなどが含まれています。

1.2 μTeaboard の起動方法

以下の手順で μTeaboard を起動してください。なお、ボードの外形やスイッチの配置などについては「ハードウェア仕様書」を参照してください。

(1) CPU ボード上の DIP-SW(DSW1) の設定を行ってください。DIP-SW の設定はかならず電源オフの状態で行ってください。

DSW1 は以下の内容です。

DSW1	1	OFF	T-Kernel を自動起動
		ON	T-Monitor を起動
	2	OFF	シリアル通信速度 38,400 bps
		ON	シリアル通信速度 57,600 bps
	3	未使用(アプリケーションで利用可)	
	4	未使用(アプリケーションで利用可)	

※ 出荷時は、上図で示した状態となっています。

SW5 (RESET) : リセットスイッチ
　システムをリセットして再起動します。ABORT 割り込みスイッチ(SW1)を押しながら起動すると、DSW1-1 OFF の状態でも、T-Mmonitor が起動します。

SW1 : ABORT 割り込みスイッチ
　割り込みを発生させて、T-Monitor に制御を移行します。T-Monitorのコマンドによりシステムを再開できます。プログラムのデバッグ用に使用します。割り込み許可状態のときのみ有効です。

SW2 : 未使用 (アプリケーションで利用可)

(2) CPU ボードのシリアルコネクタにシリアルケーブル(クロス)を使用して、開発用パソコンに接続してください。パソコン上で適当な通信ソフトウェア(製品に付属の gterm、または TeraTerm など)を起動してください。

シリアルポートの通信仕様は以下の通りです。

通信速度　　38,400 bps (または 57,600 bps)

データ長	8 ビット
ストップビット	1 ビット
パリティ	なし
フロー制御	XON/XOFF
文字コード	ASCII / EUC コード
受信行末	CR (0x0d)
送信行末	CRLF (0x0d, 0x0a)

(3) CPU ボードの AC アダプタコネクタに AC アダプタを接続してください。電源スイッチはありませんので、電源を供給するとシステムが起動して、パソコン上の通信ソフトウェアに起動メッセージが表示されます。

DSW1-1 が OFF のときは、T-Kernel が起動して、IMS のコマンド待ちの状態となります。ON のときは、T-Monitor が起動して、T-Monitor のコマンド待ちの状態となります。この場合、T-Monitor の BootDisk(BD) コマンドにより、T-Kernel を起動できます。

(4) 終了するときは、T-Monitor または IMS の EXIT コマンドにより電源をオフしてから、AC アダプタを抜いてください。

1.3 フラッシュメモリの再書き込み

出荷時には、フラッシュメモリにシステムが書き込んでありますが、再書き込みが必要になった場合は、T-Monitor のコマンドを使用して、以下の手順でフラッシュメモリに書き込んでください。

(1) DSW1-1 を ON にして、システムを起動して T-Monitor を立ち上げてください。

†ABORT 割り込みスイッチ(SW1) を押しながら、リセットスイッチ(SW5)を押しても T-Monitor が立ち上がります。

(2) FlashLoad (FLLO) コマンドを入力してください。

```
TM> FlashLoad
```

(3) 通信ソフトのファイル送信機能を使用して、CD-ROM 内の以下のファイルを無手順テキスト形式で転送してください。

soft/romimage.mot (S フォーマット形式)

†FlashLoad のコマンドを入力後、下記の行が表示された後に通信ソフトによるファイル転送を開始してください。この行が表示される前に転送を開始するとデータを取りこぼしてしまいます。

```
> Load S-Format Data of Flash ROM
```

†製品に付属している通信ソフト gterm を使用している場合は、(2)(3)の手順は、以下の１つのコマンドで行うことができます。

```
TM> .flload  soft/romimage.mot
```

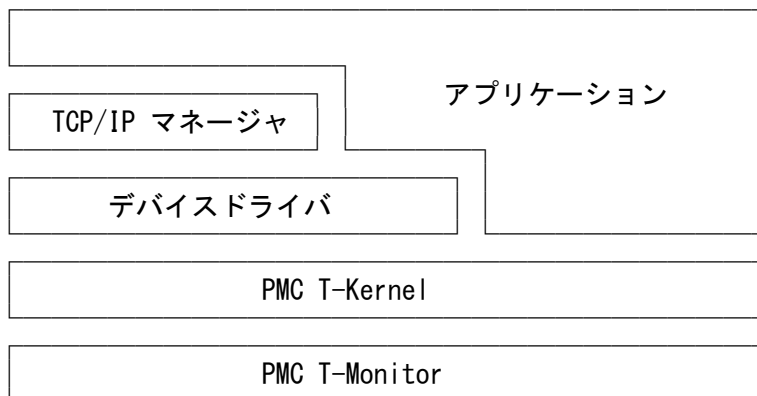
(4) 転送が終了すると、自動的にフラッシュメモリへの書き込みが行われ、システムは再度起動します。これにより、出荷時の状態に戻ります。

フラッシュメモリの内容は以下の通りです。詳細は「実装仕様書」を参照してください。

```
0x10000000 ～ 0x1000FFFF (   64  KB)  T-Monitor プログラム
0x10010000 ～ 0x1001007F (  128   B)  ROM 情報
0x10010080 ～ 0x10020000 (   63.8KB)  システム構成情報
0x10020000 ～ 0x1005FFFF (  256  KB)  T-Kernel プログラム
0x10060000 ～ 0x100FFFFF (  640  KB)  システム予約領域
0x10100000 ～ 0x103FFFFF ( 3072  KB)  ユーザ利用領域
```

2. ソフトウェア構成

2.1 全体構成

μTeaboard 全体のソフトウェア構成を以下に示します。

○ PMC T-Monitor

T-Monitor仕様書に準拠したモニタです。詳細は「T-Monitor 仕様書」および「実装仕様書」を参照してください。

○ PMC T-Kernel

T-Kernel仕様書に準拠したリアルタイムOSです。T-Kernel/OS, T-Kernel/DS, T-Kernel/SMを含んでいます。詳細は「T-Kernel 仕様書」および「実装仕様書」を参照してください。

○ デバイスドライバ

T-Kernel/SM デバイス管理機能に基づいた以下のデバイスドライバが含まれています。デバイスドライバの詳細は「デバイスドライバ説明書」および「実装仕様書」を参照してください。

時計(RTC)、コンソール(シリアル)、RS232C、LAN

○ サブシステム

T-Kernel/SM システム管理機能に基づいた以下のサブシステムが含まれています。TCP/IPマネージャの詳細は「TCP/IP マネージャ説明書」を参照してください。

TCP/IP マネージャ

○ アプリケーション

下記の開発用ツールが含まれています。

IMS (Initial Monitor System)
詳細は「IMS」の章を参照してください。

テスト/サンプルプログラムなど
開発環境に含まれるソースプログラムを参照してください。

2.2 開発対象ソフトウェアの分類

開発対象となるソフトウェアは、大きく以下の 2 つに分類され、開発方法やプログラムのオブジェクト形式などが異なってきます。

○ モニタベースのソフトウェア

T-Kernel の機能を使用しない、ハードウェア上で直接動作するソフトウェアで、T-Monitor によりプログラムをメモリ上にロードして実行します。

○ T-Kernel ベースのソフトウェア

T-Kernel の機能を使用するバイスドライバやサブシステムなどのリアルタイムソフトウェアで、常駐メモリ上で動作します。
通常、μTeaboard のアプリケーションプログラムは、T-Kernel ベースのソフトウェアとして作成します。

2.3 システムの起動手順

システムの起動は以下の手順で行われます。

(1) まず、T-Monitor が立ち上がり、ハードウェアの初期化を行います。

(2) DSW1-1 が OFF のときは、T-Monitor は、フラッシュメモリ内の ROM 情報に設定されている、Kernel 起動アドレスにジャンプします。
DSW1-1 が ON のときは、T-Monitor のコマンド入力待ちとなります。この場合は、BD コマンドにより、Kernel 起動アドレスにジャンプします。

(3) Kernel は、フラッシュメモリ内の ROM 情報に設定されている、以下のコンフィグレーションデータで指定された各種のパラメータにしたがって、初期化を行い、T-Kernel を動作させます。

SYSCONF	システムコンフィグレーション
DEVCONF	デバイスコンフィグレーション

(4) さらに、Kernel に組み込まれている以下のドライバ/サブシステムも立ち上げます。

時計(RTC), コンソール(シリアル), RS232C, LAN, TCP/IPマネージャ

(5) OS としてのすべての初期化が終了すると、フラッシュメモリ内の ROM 情報に設定されているユーザ初期化プログラムを実行します。ユーザ初期化プログラムが定義されていない場合は、デフ

ォールトの初期タスクとして IMS を立ち上げて、IMS のコマンド入力待ちとなります。

(6) ユーザ初期化プログラムでは、ドライバやサブシステムなどを起動し、さらにアプリケーションタスクを起動します。実行後は、その戻り値により、IMS を起動する、電源オフ、または、再起動する、のいずれかの処理となります。

3. ソフトウェア開発方法

GNU 開発環境のインストールやコンパイル方法に関しては、「GNU開発環境説明書」を参照してください。

3.1 モニタベースのソフトウェア

モニタベースのソフトウェアは、モニタで使用している資源を除いてすべてのハードウェア資源を利用することができます。

モニタでは、0x20000000 ～ 0x20001FFF までの 8 KB の RAM 領域を使用していますので、この領域を除いたアドレスにプログラムをリンクするようにしてください。

作成したプログラムは、S フォーマット形式に変換して、モニタの Load コマンドによりメモリ上にロードして、Go コマンドで実行します。

3.2 T-Kernel ベースのソフトウェア

デバイスドライバ、サブシステムやアプリケーションタスクなどの T-Kernel ベースのソフトウェアは、RAM または、フラッシュメモリ上のユーザ領域で動作します。T-Kernel では、プログラムのメモリ領域の管理は行いません。

ユーザ領域	アドレス範囲	
フラッシュメモリ	(0x10060000 ～ 0x100FFFFF)	640 KB (*1)
	0x10100000 ～ 0x103FFFFF	3 MB (*2)
RAM	0x20100000 ～ 0x201FFFFF	1 MB (*3)
内蔵 SRAM	0x00000200 ～ 0x00001FFF	7.5 KB (*4)

†フラッシュメモリは (*2) の領域を使用しますが、不足する場合は、(*1)のシステム予約領域も利用可能です。ただし、今後の OS のバージョンアップにより、開始アドレス(0x10060000)が変更されることがあります。

†RAM 領域は、T-Kernel が管理する領域(0x20000000 ～ 0x200FFFFF)と管理しないユーザ領域(*3)に 2分割され、その境界アドレス (0x20100000) は、ROM 情報の設定により変更可能です。
T-Kernel が管理する領域は、タスクのスタックや動的に確保されるメモリに使用されますので、システムの動作に必要なサイズを確保してください。IMS の ref_smb コマンドにより、実際に使用されているサイズを知ることができます。

†内蔵 SRAM 領域(*4)の直前の 512 B (0x00000000 ～ 0x000001FF) は、例外分岐処理に使用されていますので、破壊しないように気を付けてください。

作成したプログラムは、S フォーマット形式に変換して、モニタの Load コマンドによりメモリ(RAM)上にロードして、IMS の lodspg コマンドで、アドレスを指定して実行を開始します。

フラッシュメモリ上で動作するプログラムは、モニタの FlashLoad コマンドで直接フラッシュメモ

リに書き込むか、一旦 RAM 上のロードしてから WriteROM コマンドでフラッシュメモリ書き込みます。フラッシュメモリ上のプログラムも、IMS の lodspg コマンドでアドレスを指定して実行を開始します。

3.3 ユーザ初期化プログラム

デバイスドライバ、サブシステムやアプリケーションタスクなどを、システムに組み込んで自動的に実行させる場合は、それらを起動するためのユーザ初期化プログラムを作成する必要があります。

ユーザ初期化プログラムは、フラッシュメモリ上において、その実行開始アドレスを ROM 情報に設定します。

ユーザ初期化プログラムのサンプルは、開発環境の $(BD)/kappl/userinit/* にありますので参照してください。

4. ネットワーク機能

μTeaboard では、BSD ソケットI/F に準拠した API を使用して、TCP/IP 通信を行うことができます。

API の詳細に関しては「TCP/IPマネージャ説明書」を参照してください。

4.1 LAN ドライバ設定

ネットワークを使用するためには、フラッシュメモリ内のシステム構成情報に(DEVCONF)内に、LAN ドライバに関する以下の設定を行う必要があります。

```
NETDEV0 kind addr irq

kind    使用するネットワークコントローラ
addr    I/O アドレス
irq     割り込みベクタ番号
```

デフォールトで以下の設定が行われており、この設定のみ有効です。

```
NETDEV0 0x0011 0x40000000 56
```

なお、kind の値に以下の値を OR することで、ネットワークコントローラの動作を変更することができます。

```
0x00800 常時電源 ON
0x02000 10Mbps 固定
0x04000 100Mbps 固定
0x10000 Full-Duplex 指定
```

†常時電源 ON を指定しない場合、通信時のみコントローラの電源が ON となります。
†通信速度を指定しない場合、Auto-negotiation (10Mbps/100Mbps 自動切替)で動作します。
†Full-Duplex を指定しない場合、Half-Duplex で動作します。

システム構成情報(DEVCONF)の変更は、開発環境の $(BD)/kernel/config/tbat91 ディレクトリにある DEVCONF ファイルを編集した後、rominfo.mot ファイルを作成して、フラッシュメモリに書き込んでください。

4.2 ネットワーク環境設定

ホスト IP アドレスや、DNS サーバ IP アドレスなどのネットワーク環境は、ネットワークの使用開始時にプログラムで設定する必要があります。

ホスト IP アドレスを設定しない（IP アドレス = 0.0.0.0）場合は、DHCP サーバから IP アドレスを自動取得しますが、ケーブル未接続などの原因で、自動取得に失敗した場合は、一度ネットワークの使用を終了しないと、再度、DHCP サーバからの取得は行われません。

開発環境の $(BD)/kappl/nettool/* に、ネットワークを使用するためのサンプルプログラム(nettool)がありますので参考にしてください。

CD-ROM 内のソフトウェアに、RAM 上で動作する nettool の実行ファイル nettool.mot が入っていますので、以下の手順で実行してください。

```
[IMS]% #
        -- プログラムロードのため T-Monitor に入る
TM> .load /usr/local/te/kappl/nettool/tbat91/nettool.mot
        -- nettool.mot をロードする
.. LO XS [/usr/local/te/kappl/nettool/tbat91/nettool.mot: 82106 bytes]
....*....*....*....*....*....*....*....* (4561.4 bytes/sec)

Loaded: 20100000 -> 20108033
TM> g
        -- T-Monitor を抜ける
[IMS]% lodspg @20100000
        -- ロードした nettool を実行する
nettool - start
? start 192.169.0.10 192.168.0.1 test.jp
-- ホストIPアドレス, DNS アドレス, ドメイン名を指定してネットワークの使用を開始す
   る
? ping testhost1
-- ping を実行する
testhost1 is alive <192.168.0.20> : 10 ms
? end
-- ネットワークの使用を終了する
? start
-- ホストIPアドレスを指定せずに(DHCP 利用)ネットワークの使用を開始する
? ping 192.168.0.20
-- ping を実行する
192.168.0.20 is alive <192.168.0.20> : 40 ms
? end
-- ネットワークの使用を終了する
? q
-- nettool を終了する
nettool - end
SYSPRG @20100000 [0]
[IMS]%
```

なお、$(BD)/kappl/netool/src/ にある netio.h, netio.c は、サンプルプログラム nettool では使用していませんが、TCP/IP マネージャを使用するための上位関数群ですので、プログラミングの参考にしてください。

5. IMS

5.1 概要

IMS (Initial Monitor System) は、T-Kernel の初期タスクとして起動されるプログラムであり、以下の機能を持っている。

・コマンドによる T-Kernel に各種状態の参照/操作
・システムプログラム(サブシステムやデバイスドライバ)の起動/終了

コマンド入力待ちの状態では、コンソールに次のプロンプトを表示する。

[IMS]%

コマンドの書式は次のようになる。

・ ＊ で始まる行はコメントとして無視される。
・コマンドの中には、数値引数や文字列引数をとるものがある。
・文字列は ″ ″ でくくって入力する。
・数値定数は次のように入力する。
  ・H'に続く16進数字列(0～9, A～F) または 0xに続く16進数字列(0～9, A～F)
  ・D'に続く10進数字列(0～9)
  ・Q'に続く8進数字列(0～7)
  ・B'に続く2進数字列(0～1)
  ・単なる数字列の場合は radix コマンドで設定した基数に従って解釈する。
・数値は+-*/により加減乗除ができる。除算の場合は、小数点以下は切捨てられる。演算子の優先順位はすべて等しく、左から順に解釈される。例えば -8/3 は -2、1+2*3 は 9 と解釈される(10進数表現の場合)。

5.2 IMS コマンド一覧

---

h
コマンドヘルプ

【形式】 h [<コマンド名>]

指定したコマンドのヘルプを表示する。

引数を省略するとコマンド一覧を表示する。

?
コマンドヘルプ

【形式】 ? [<コマンド名>]

hコマンドと同じである。

---

prompt
プロンプト設定

【形式】 prompt ″文字列″

″文字列″をプロンプトに設定する。″文字列″は最大32文字までである。

---

radix
基数の設定

【形式】 radix [{ H | D | O }]

入力に用いる数値の基数を、H(16進数)、D(10進数)、O(8進数)のいずれかに設定する。

引数を省略すると現在設定されている基数を表示する。

---

sleep
スリープ

【形式】 sleep ［〈タイムアウト時間〉］

IMSを待ち状態に移行させる。[CTRL]+[G]キーの入力で待ち解除される。またタイムアウト時間(ミリ秒単位)を指定した場合は、その時間が経過すると待ち解除される。

---

ref_tsk
タスクの状態参照

【形式】 ref_tsk ［〈タスクID〉］

タスクIDで指定したタスクについて、次の情報を表示する。引数を省略した場合は、すべてのタスクについて表示する。

- TID　タスクID
- PRI:BPR　現在優先度、ベース優先度
- SLT　スライスタイム
- WUP　起床要求数
- SUS　強制待ち要求数
- STS　タスク状態（待ち状態の場合は待ち要因オブジェクトも）
- ST+UT　システム実行時間、ユーザ実行時間
- RID　リソースID
- EXINF/NAME　拡張情報

---

ref_sem
セマフォの状態参照

【形式】 ref_sem ［〈セマフォID〉］

セマフォIDで指定したセマフォについて、次の情報を表示する。引数を省略した場合は、すべてのセマフォについて表示する。

- ID　セマフォID
- WID　先頭の待ちタスクのID（待ちタスクがない場合は0）
- CNT　現在のセマフォカウント値
- EXINF　拡張情報

---

ref_flg
イベントフラグの状態参照

【形式】 ref_flg [〈イベントフラグID〉]

　イベントフラグIDで指定したイベントフラグについて、次の情報を表示する。引数を省略した場合は、すべてのイベントフラグについて表示する。

・ ID　　　イベントフラグID
・ WID　　　先頭の待ちタスクのID (待ちタスクがない場合は0)
・ PTN　　　現在のフラグパターン
・ EXINF　　拡張情報

---

ref_mbx
メールボックスの状態参照

【形式】 ref_mbx [〈メールボックスID〉]

　メールボックスIDで指定したメールボックスについて、次の情報を表示する。引数を省略した場合は、すべてのメールボックスについて表示する。

・ ID　　　メールボックスID
・ WID　　　先頭の待ちタスクのID (待ちタスクがない場合は0)
・ MSG　　　先頭のメッセージのアドレス (メッセージがない場合は0)
・ EXINF　　拡張情報

---

ref_mtx
ミューテックスの状態参照

【形式】 ref_mtx [〈ミューテックスID〉]

　ミューテックスIDで指定したミューテックスについて、次の情報を表示する。引数を省略した場合は、すべてのミューテックスについて表示する。

・ ID　　　ミューテックスID
・ HID　　　ロックしているタスクのID (ロックされていない場合は0)
・ WID　　　先頭の待ちタスクのID (待ちタスクがない場合は0)
・ EXINF　　拡張情報

---

ref_mbf
メッセージバッファの状態参照

【形式】 ref_mbf [〈メッセージバッファID〉]

　メッセージバッファIDで指定したメッセージバッファについて、次の情報を表示する。引数を省略した場合は、すべてのメッセージバッファについて表示する。

・ ID　　　メッセージバッファID
・ WID　　　先頭の受信待ちタスクのID（受信待ちタスクがない場合は0）
・ SID　　　先頭の送信待ちタスクのID（送信待ちタスクがない場合は0）
・ MSGSZ　　先頭のメッセージのサイズ（メッセージがない場合は0）
・ FREE　　空きバッファのサイズ
・ MAX　　　メッセージバッファ生成時に指定したメッセージ最大長
・ EXINF　　拡張情報

---

ref_por
ランデブポートの状態参照

【形式】 ref_por [〈ランデブポートID〉]

　ランデブポートIDで指定したランデブポートについて、次の情報を表示する。引数を省略した場合は、すべてのランデブポートについて表示する。

・ ID　　　ランデブポートID
・ WID　　　先頭の呼出待ちタスクのID（呼出待ちタスクがない場合は0）
・ AID　　　先頭の受付待ちタスクのID（受付待ちタスクがない場合は0）
・ MAXCSZ　　ランデブポート生成時に指定した呼出時のメッセージの最大長
・ MAXRSZ　　ランデブポート生成時に指定した返答時のメッセージの最大長
・ EXINF　　拡張情報

---

ref_mpl
可変長メモリプールの状態参照

【形式】 ref_mpl [〈可変長メモリプールID〉]

　可変長メモリプールIDで指定した可変長メモリプールについて、次の情報を表示する。引数を省略した場合は、すべての可変長メモリプールについて表示する。

・ ID　　　可変長メモリプールID
・ WID　　　先頭の待ちタスクのID（待ちタスクがない場合は0）
・ FREE　　空き領域の合計サイズ
・ MAX　　　最大の空き領域のサイズ

・ EXINF 拡張情報

---

ref_mpf
固定長メモリプールの状態参照

【形式】 ref_mpf [<固定長メモリプールID>]

固定長メモリプールIDで指定した固定長メモリプールについて、次の情報を表示する。引数を省略した場合は、すべての固定長メモリプールについて表示する。

・ ID 固定長メモリプールID
・ WID 先頭の待ちタスクのID（待ちタスクがない場合は0）
・ FREE 空きブロックの個数
・ EXINF 拡張情報

---

ref_cyc
周期ハンドラの状態参照

【形式】 ref_cyc [<周期ハンドラID>]

周期ハンドラIDで指定した周期ハンドラについて、次の情報を表示する。引数を省略した場合は、すべての周期ハンドラについて表示する。

・ ID 周期ハンドラID
・ STS 周期ハンドラの状態（STA=動作中）
・ TIME 次のハンドラ起動までの残り時間
・ EXINF 拡張情報

---

ref_alm
アラームハンドラの状態参照

【形式】 ref_alm [<アラームハンドラID>]

アラームハンドラIDで指定したアラームハンドラについて、次の情報を表示する。引数を省略した場合は、すべてのアラームハンドラについて表示する。

・ ID アラームハンドラID
・ STS アラームハンドラの状態（STA=動作中）
・ TIME ハンドラ起動までの残り時間
・ EXINF 拡張情報

---

get_reg
タスクレジスタの参照

【形式】 get_reg <タスクID>

タスクIDで指定したタスクのタスクレジスタの内容を表示する。

---

sta_tsk
タスクの起動

【形式】 sta_tsk <タスクID> [, <タスク起動コード>]

タスクIDで指定したタスクを起動する。

休止状態(DORMANT)でないタスクは起動できない。

---

ter_tsk
タスクの強制終了

【形式】 ter_tsk <タスクID>

タスクIDで指定したタスクを強制終了する。

---

del_tsk
タスクの削除

【形式】 del_tsk <タスクID>

タスクIDで指定したタスクを削除する。

休止状態(DORMANT)でないタスクは削除できない。

---

rel_wai
タスクの待ち状態解除

【形式】 rel_tsk <タスクID>

タスクIDで指定したタスクの待ち状態を解除する。

待ち状態(WAITING)でないタスクは待ち解除できない。

---

wup_tsk
タスクを起床

【形式】 wup_tsk <タスクID>

タスクIDで指定したタスクを起床する。

対象タスクがtk_slp_tskを実行しておらず、待ち状態でない場合には、起床要求はキューイングされる。

---

can_wup
タスクの起床要求を無効化

【形式】 can_wup <タスクID>

タスクIDで指定したタスクの起床要求を無効化する。すなわち起床要求をすべてキャンセルする。

---

sus_tsk
タスクを強制待ち状態へ移行

【形式】 sus_tsk <タスクID>

タスクIDで指定したタスクを強制待ち状態(SUSPENDED)へ移行させる。

---

rsm_tsk
強制待ち状態のタスクを再開

【形式】 rsm_tsk <タスクID>

タスクIDで指定したタスクの強制待ち状態(SUSPENDED)を解除する。強制待ち要求のネストを1回分だけ解除する。

---

frsm_tsk
強制待ち状態のタスクを強制再開

【形式】 frsm_tsk 〈タスクID〉

タスクIDで指定したタスクの強制待ち状態(SUSPENDED) を強制解除する。強制待ち要求のネストをすべて解除する。

---

chg_pri
タスクの優先度設定

【形式】 chg_pri 〈タスクID〉, 〈優先度〉

タスクIDで指定したタスクの優先度を設定する。

---

rot_rdq
タスクの優先順位の回転

【形式】 rot_rdq 〈優先度〉

指定された優先度のタスクの優先順位を回転する。

---

sig_sem
セマフォ資源返却

【形式】 sig_sem 〈セマフォID〉, 〈増分〉

セマフォIDで指定したセマフォのセマフォカウントを指定した増分だけ増やす。

---

clr_flg
イベントフラグのクリア

【形式】 clr_flg 〈イベントフラグID〉, 〈クリアパターン〉

イベントフラグIDで指定したイベントフラグをクリアパターン指定によってクリアする。

---

set_flg
イベントフラグのセット

【形式】 set_flg <イベントフラグID>, <セットパターン>

イベントフラグIDで指定したイベントフラグをセットパターン指定によってセットする。

---

set_pow
省電力モード設定

【形式】 set_pow <省電力モード>

省電力モードを設定する。

1	サスペンド状態へ移行
2	低消費電力モード切替禁止
3	低消費電力モード切替許可

---

ref_smb
システムメモリ状態参照

【形式】 ref_smb

システムメモリ状態を参照する。

Blksz=	1メモリブロックのバイト数
Total=	全体のブロック数
Free=	未使用ブロック数
[XX% used]	使用ブロック数の比率

---

exit
システム停止

【形式】 exit [<引数>]

システムを停止する。

引数なし	シャットダウン（電源オフ）
-1	初期スタート（Reset）

機種によっては、自動的には電源が切れない場合がある。

---

lodspg
カーネル拡張モジュールの起動

【形式】 lodspg @<アドレス> <パラメータ>...

指定されたアドレスから、拡張モジュールとして起動(実行)する。

---

unlspg
カーネル拡張モジュールの終了

【形式】 unlspg @<アドレス>

指定されたアドレスから、終了処理として拡張モジュールを実行する。

#
T-Monitorインタフェース

【形式】 # [<T-Monitorコマンド>]

引数として指定されたT-Monitorコマンドを実行する。

引数を省略すると、T-Monitorに入る。T-MonitorのGコマンドを使えばIMSに戻る。

---

date
日付表示/設定

【形式】 date [<year>,<month>,<day> [,<hour>,<min>,<sec>]]

現在の日付を表示/設定する。引数を省略すると表示する。

5.3 IMS コマンド索引